

AMPLIACIÓN DE LA BBDD PARA GUARDAR DISTINTAS DIETAS

NOTA: en caso de aumentar con otro tipo de dietas (baja en calorías, vegetariana, vegana, diabéticos, etc ...), incluiremos un atributo enumerado "Categoría" a la Receta, para recoger esta característica.

ATRIBUTOS ENUMERADOS:

Dieta:

NomDieta --> Diaria, Fin de semana, Semanal

Menu:

TipoComida --> Desayuno, Almuerzo, Comida, Merienda, Cena

contiene:

Dia --> Lunes, Martes, Miércoles, Jueves, Viernes, Sábado, Domingo

Receta:

Tiempo --> < 15 min., 15 min., > 30 min., > 30 min.

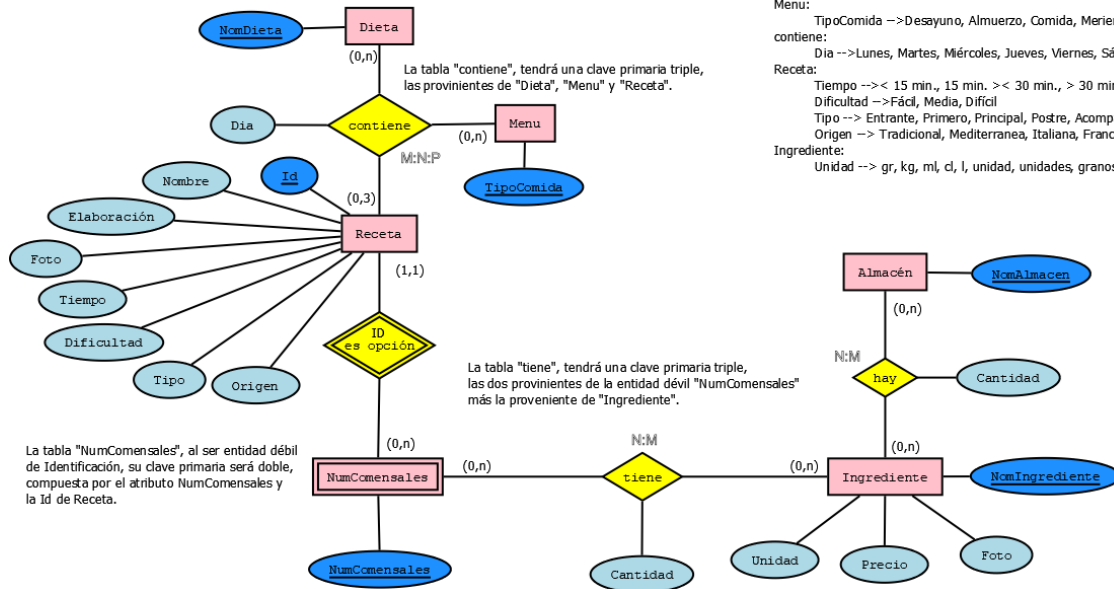
Dificultad --> Fácil, Media, Difícil

Tipo --> Entrante, Primer plato, Principal, Postre, Acompañamiento, Pincho, Bebida

Origen --> Tradicional, Mediterránea, Italiana, Francesa, China, Japonesa, India, Mexicana, ...

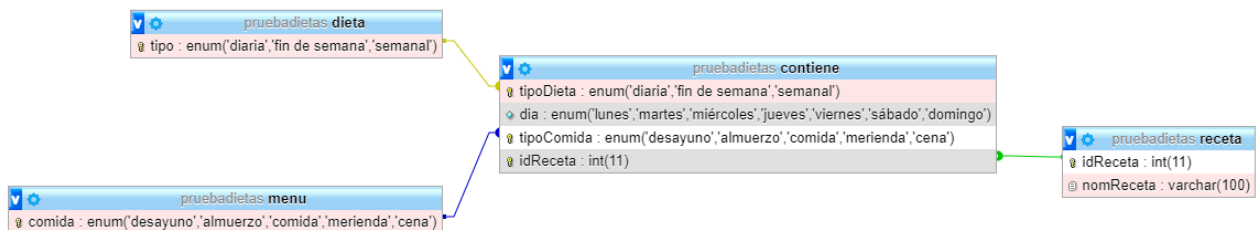
Ingrediente:

Unidad --> gr, kg, ml, cl, l, unidad, unidades, granos, cucharita, cuchara, ...



Realizo prueba en MySQL para comprobar funcionamiento (en Receta sólo uso Id y Nombre):

El diseño



El Script

```
create database pruebaDietas collate utf8mb4_spanish_ci;
```

```
use pruebaDietas;
```

```
create table RECETA
```

```
(idReceta int AUTO_INCREMENT primary key,
```

```
nomReceta varchar(100) not null);
```

```
create table MENU
```

```
(comida enum('desayuno','almuerzo','comida','merienda','cena') primary key);
```

```
create table DIETA
```

```
(tipo enum('diaria','fin de semana','semanal') primary key);
```

```
create table CONTIENE
```

```
(tipoDieta enum('diaria','fin de semana','semanal'),
```

```
dia enum('lunes','martes','miércoles','jueves','viernes','sábado','domingo'),
```

```
tipoComida enum('desayuno','almuerzo','comida','merienda','cena'),
```

```
idReceta int,
```

```
constraint fk_idReceta foreign key(idReceta) references RECETA(idReceta) on delete cascade on update cascade,
```

```
constraint fk_tipoComida foreign key(tipoComida) references MENU(comida) on delete cascade on update cascade,
```

```
constraint fk_tipoDieta foreign key(tipoDieta) references DIETA(tipo) on delete cascade on update cascade,
```

```
primary key(idReceta,tipoComida,tipoDieta));
```

```
insert into RECETA values (1,'Café con leche');
```

```
insert into RECETA values (2,'Pincho de tortilla');
```

```
insert into RECETA values (3,'Arroz a la cubana');
```

```
insert into RECETA values (4,'Filete de ternera con patatas');
```

```
insert into RECETA values (5,'Arroz con leche');
```

```
insert into RECETA values (6,'Té con pastas');
```

```
insert into RECETA values (7,'Tortilla francesa');
```

```
insert into RECETA values (8,'Yogur natural');
```

```
insert into MENU values ('desayuno');
```

```
insert into MENU values ('almuerzo');
```

```
insert into MENU values ('comida');
```

```
insert into MENU values ('merienda');
```

```
insert into MENU values ('cena');
```

```
insert into DIETA values ('diaria');
```

```
insert into DIETA values ('fin de semana');
```

```
insert into DIETA values ('semanal');
```

```
insert into contiene values ('diaria','lunes','desayuno',1);
```

```
insert into contiene values ('diaria','lunes','almuerzo',2);
```

```
insert into contiene values ('diaria','lunes','comida',3);
```

```
insert into contiene values ('diaria','lunes','comida',4);
```

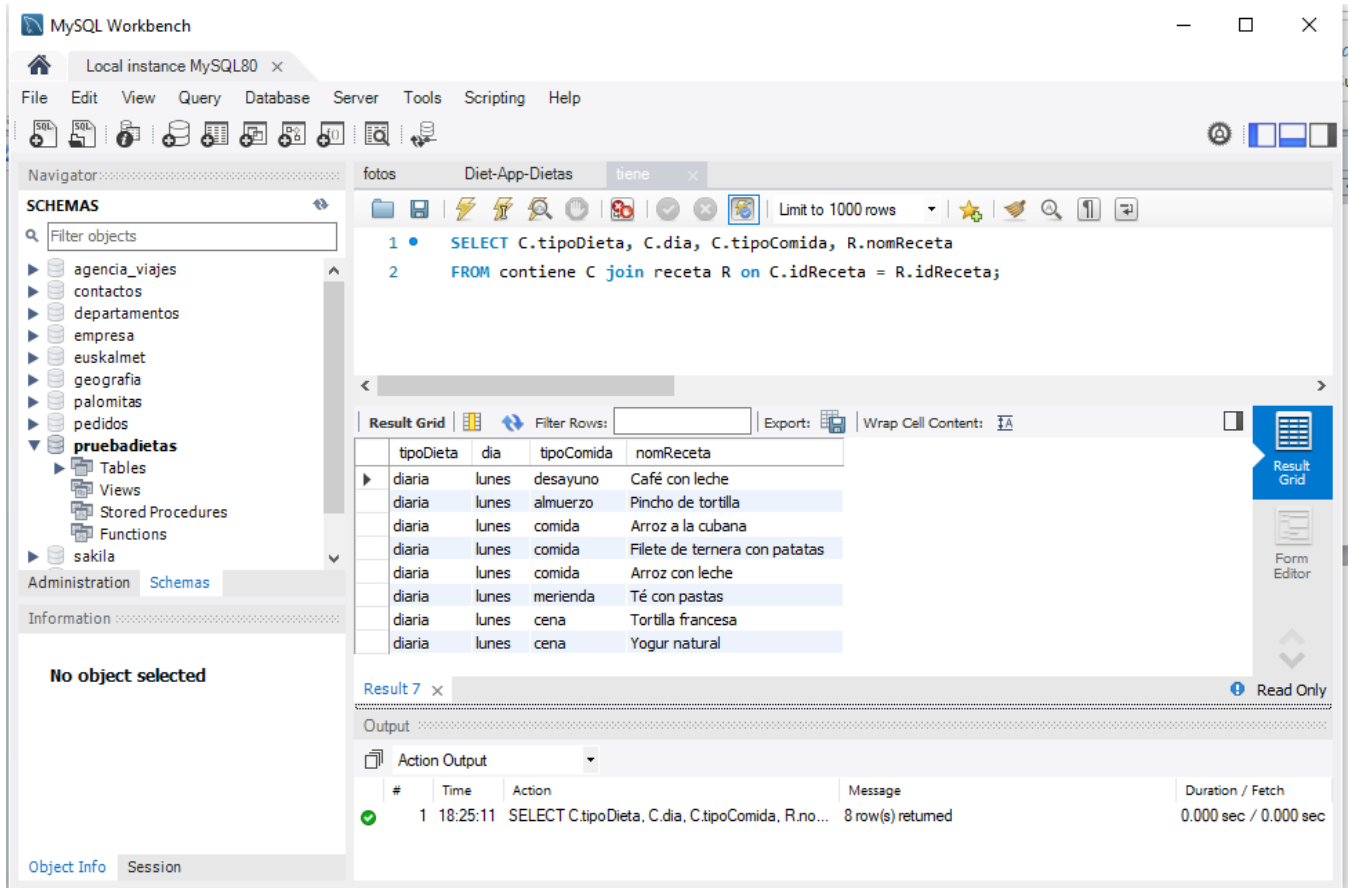
```
insert into contiene values ('diaria','lunes','comida',5);
```

```
insert into contiene values ('diaria','lunes','merienda',6);
```

```
insert into contiene values ('diaria','lunes','cena',7);
```

```
insert into contiene values ('diaria','lunes','cena',8);
```

Ejecución de consulta



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'pruebadietas' selected. The main window shows a query editor with the following SQL query:

```
1 • SELECT C.tipoDieta, C.dia, C.tipoComida, R.nomReceta
2 FROM contiene C join receta R on C.idReceta = R.idReceta;
```

The 'Result Grid' tab is active, displaying the following data:

tipoDieta	dia	tipoComida	nomReceta
diaria	lunes	desayuno	Café con leche
diaria	lunes	almuerzo	Pincho de tortilla
diaria	lunes	comida	Arroz a la cubana
diaria	lunes	comida	Filete de ternera con patatas
diaria	lunes	comida	Arroz con leche
diaria	lunes	merienda	Té con pastas
diaria	lunes	cena	Tortilla francesa
diaria	lunes	cena	Yogur natural

The bottom panel shows the 'Output' tab with the following message:

#	Time	Action	Message	Duration / Fetch
1	18:25:11	SELECT C.tipoDieta, C.dia, C.tipoComida, R.no...	8 row(s) returned	0.000 sec / 0.000 sec

NOTA: Los tipos de datos ENUM no existen en SQLite, pero se soluciona utilizando la restricción CHECK, como se explica en el siguiente enlace:

[How to create ENUM type in SQLite? - Stack Overflow](#)

SQLite way is to use a [CHECK constraint](#).

Some examples:

```
CREATE TABLE prices (
  id INTEGER PRIMARY KEY,
  pName TEXT CHECK( LENGTH(pName) <= 100 ) NOT NULL DEFAULT '',
  pType TEXT CHECK( pType IN ('M','R','H') ) NOT NULL DEFAULT 'M',
  pField TEXT CHECK( LENGTH(pField) <= 50 ) NULL DEFAULT NULL,
  pFieldExt TEXT CHECK( LENGTH(pFieldExt) <= 50 ) NULL DEFAULT NULL,
  cmp_id INTEGER NOT NULL DEFAULT '0'
)
```

This will limit the pType column to just the values M, R, and H, just like enum("M", "R", "H") would do in some other SQL engines.



edited Mar 14 '19 at 22:38

[Martin Tournioj](#)

23.3k2323 gold badges8989 silver badges122122 bronze badges